BDD and Pre-computation based strategy for Power optimization of a 4-bit Comparator

Satish.Ajjappanavar[*1],Dr.AnilkumarV.Nandi[#2]

**Dept of Electronics & communication*
BVB College of Engineering and Technology
Hubli,India

[#]Professor ,Dept of Electronics &communication
BVB College of Engineering and Technology
Hubli,India

[1]satish.ajjappanavar@gmail.com

[2]anilnandy@bvb.edu

Abstract : **BDD and Precomputation strategies are the general evaluation methodologies for symbolic model checking, In BDD based realization of logic circuits, the area and power consumption is determined by the total number of nodes. A proper polarity selection of the sub-functions can not only reduce the number of BDD nodes, but also the switching activity,Precomputation is the method of "input subset aborting" technique so that we can optimize the power by reducing the switching activity in the next clock signal,The Cadence SOC encounter tool which provides the power levels of the each Design for low power, This study addresses the performance issue of 4-bit magnitude comparator specially for low power.**

Key words:BDD,Precomptation,Cadence SOC Encounter tool

1.INTRODUCTION

Today, digital design, almost without exception, uses synthesis tools at the register transfer level, which require a designer to explicitly incorporate power reducing features within the system design. Approaches exist to lowering the power. One is to reduce the operating voltage of the circuit, or to reduce the voltage supply. The weight and cost of power supply generally depends on the maximum possible power used at some instant. These are guided by a global cost function, which evaluates the current configuration with respect to user-specified objectives on area, delay, and now power consumption. The current trend towards low-power design is mainly driven by two forces the growing demand for long-life autonomous portable equipment, and the technological limitations of high-performance VLSI systems.For the first category of products, low-power is the major goal for which speed and dynamic range might have to be sacrificed. High speed and high integration density are the objectives for the second application category. The most efficient way to reduce the power consumption of digital circuits is to reduce the supply voltage. Power optimization approaches at the High-level are significant since research results Indicates that higher levels of abstraction have greater potential for power reduction.

2.BDD INTRODUCTION

Binary Decision Diagrams (BDDs) are useful data structures for symbolic Boolean manipulations. For the last two decades, BDD have gained great popularity in representing discrete functions. BDD is graph representation of Boolean functions proposed by Bryant and Akers [3]. It is a directed acyclic graph; the graph has two sink nodes labelled 0 and 1 representing the Boolean functions 0 and 1. Each sink node is labelled with a Boolean variable and has two out edges labelled 1 and 0. BDDs are used in many tasks in VLSI, such as equivalence checking, property checking, logic synthesis, and false paths. In this paper we describe a new approach for the realization of a BDD package, to perform manipulations of Boolean functions. Synthesis, verification, and testing algorithms of VLSI circuits manipulate large number of switching functions. Therefore it is important to have efficient methods to represent and manipulate such functions.

Consider the switching function,
f= AvBC

and assume we are interested in defining a procedure for determining the binary value of f given the binary values of A, B, and C. One way to do this would be to begin by looking at the value of A. If A = 1, then f= 1 and we are finished. If A = 0, we look at B. If B = 1, then f= 0 and again we are finished. Otherwise, we look at C and its value will be the value of f Fig 2.1 shows a simple diagram of this procedure.

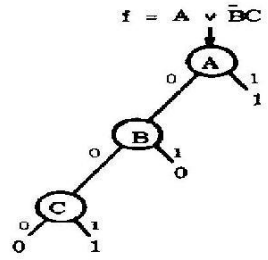

Figure 2.1: BDD representation of function A+B'C [3]

Binary Decision diagram based minimization of a logic circuit plays a significant role. A Boolean function can be in factored form in multi level realization. BDD is a form of realization of multi level logic and is resulted from direct implementation of Shannon's Expansion of a logic function. Each node in a BDD signifies some logic function and a decision is made at each node of it and determines either the node is getting a zero value or one at each leg. This continues until the tree diagram reaches its leaf, Each BDD node can be easily realized using multiplexer.Thus, minimization of total number of nodes of a BDD means minimizing the number of variable present in th

function, hence reduction in area [3]. A proper polarity selection of the sub-functions can reduce not only the number of BDD nodes, but also the switching Consider the diagram in Fig. 2.2(a) which results from the truth table for f= ABC v AC. We note that the value of f obtained at the leftmost C-node is 0 regardless of the value of C. Accordingly; we can remove this node and replace it by 0. Likewise, the two rightmost C-nodes are identical in the sense that they lead to identical output values, so we can combine them into a single node. The result is Fig. 2.2(b).

But now we note that the rightmost B-node is superfluous, since both of its branches go to the same node. Thus, we can remove it to obtain the simplified diagram of Fig. 2.2(c).

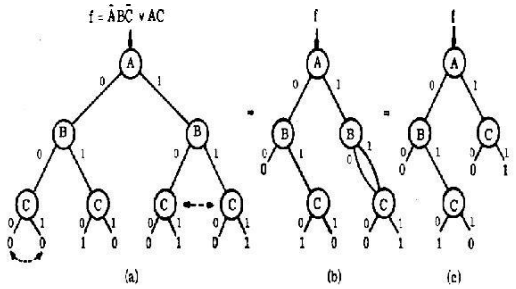

Figure 2.2: Optimization of BDD of function f [3]

3. BDD PACKAGE [2]

BDD packages typically share common implementation features. There are three main components in a BDD package: the BDD algorithm component, the dynamic variable reordering component, and the garbage collection component. In this section, we describe the common features in each of these components.

3.1 BDD Algorithm

This component computes the result BDDs for various Boolean operations. The implementation of these algorithms is typically based on depth-first traversal. The unique tables are hash tables with the hash collisions resolved by chaining. A separate unique table is associated with each variable to facilitate the dynamic-variable-reordering process. The computed cache is a hash-based direct mapped (1-way associative) cache.

BDD nodes support complement edges where for each edge, an extra bit is used to indicate whether or not the target function should be complemented (Boolean negation). The advantage of this encoding is that a function and its complement can be represented by the same BDD and use this extra bit in the reference edge to interpret the BDD either in the positive or the negated form. Implementation-wise, this extra bit is typically encoded in the least significant bit of the address pointer (the reference edge) to avoid incurring extra memory cost. This encoding exploits the property that address pointers in modern machines are always at least 4-byte aligned, which means the least significant bit is always 0. Thus it can be used to encode the complement information.

3.2 Dynamic Variable Reordering

As the variable order can have significant impact on the size of a BDD graph, dynamic variable reordering is an essential part of all modern BDD packages. The goal for this component is to dynamically establish a good variable order as the computation progresses. Typically, when a variable reordering algorithm is invoked, all top-level operations that are currently being processed are aborted. When the variable reordering algorithm terminates, these aborted operations are restarted from the beginning. The dynamic variable reordering algorithms are generally based on the *sifting* algorithm; i.e., the variable orders are changed by exchanging nodes in one level with nodes in the adjacent level. Figure 3.2.1 illustrates the sifting process for the 2-bit comparator example.

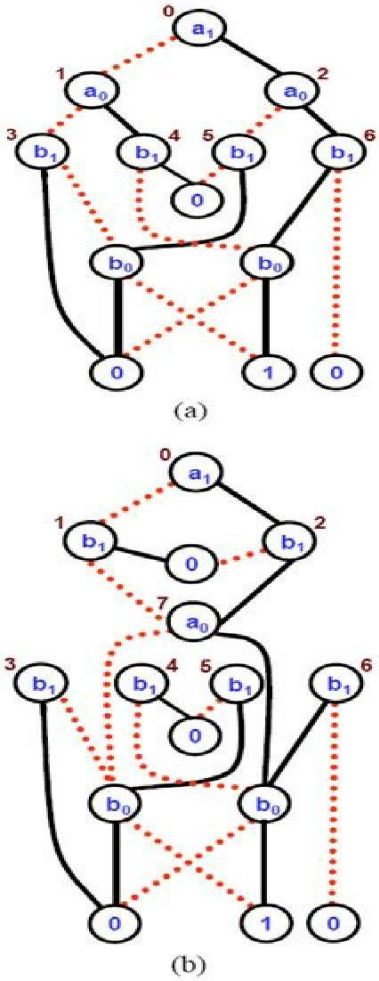

Figure 3.2.1: Shifting process for 2-bit comparator [2]

3.3 Garbage Collection

BDD computations are inherently memory intensive because after all, it is all about traversing and constructing graphs. Furthermore, in verification, many intermediate BDD results are created to arrive at a simple final answer—true or false. Thus, it is important to have a good garbage collector to automatically remove BDD nodes that are no longer useful. We will refer to a BDD node as *reachable* if it is in some BDD that external user has a reference to. As external users free references to BDDs, some BDD nodes may no longer be reachable (*deaths*). We will refer to these nodes as *unreachable* BDD nodes.

4.MAGNITUDE COMPARATOR

Comparing two binary numbers for equality is a commonly used operation in computer system and device interfaces. A circuit that compares two binary numbers and indicates whether they are equal is called a comparator. Comparators interpret their input numbers as signed or unsigned numbers and also indicate an arithmetic relationship between the numbers often called magnitude comparator.

The logic diagram of 4-bit magnitude comparator is shown in fig 4.1 [11]. It provides greater-than output A > B and less-than output A < B as well as an equal output A = B. In normal operation exactly one input and one output should be asserted.The logic for a 4-bit magnitude comparator is as follows:

Let the two 4-bit numbers be A= $A_3A_2A_1A_0$ and B = B_3 B_2 B_1 B_0.

1. If $A_3 = 1$ and $B_3 = 0$, then A > B. Or
2. If A_3 and B_3 coincide, and if $A_2 = 1$ and $B_2 = 0$, then A > B. Or
3. If A_3 and B_3 coincide, and if A_2 and B_2 coincide, and if $A_1 = 1$ and $B_1 = 0$, then A > B.

If A_3 and B_3 coincide, and if A_2 and B_2 coincide, if A_1 and B_1 coincide, and if $A_0 = 1$ and $B_0 = 0$, then A > B.

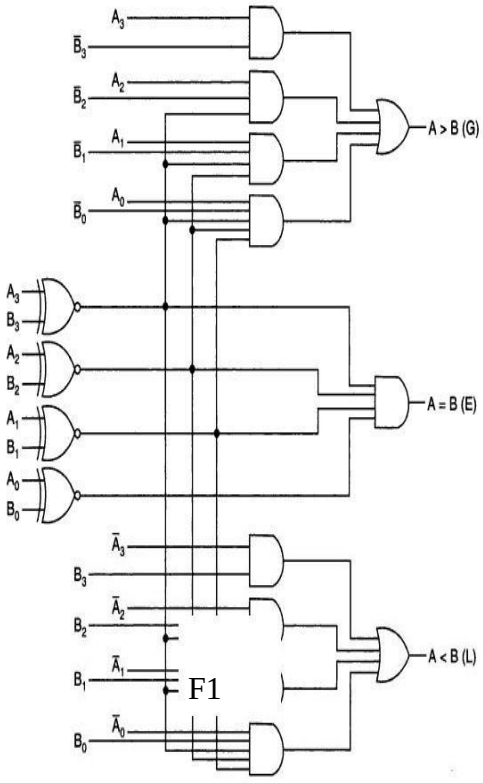

Figure 4.1: Logic diagram of 4-bit comparator [11]

5.PRE-COMPUTATION

We present a powerful sequential logic optimization method that is based on selectively precomputing the output logic values of the circuit one clock cycle before they are required, and using the precomputed values to reduce internal switching activity in the succeeding clock cycle. The primary optimization step is the synthesis of precomputation logic, which computes the output values for a subset of a input conditions. If the output values can be precomputed, the original logic circuit can be "turned off" in the next clock cycle and will not have any switching activity. Since the savings in the power dissipation of the original circuit is offset by the power dissipated in the precomputation phase, the selection of the subset of input conditions for which the output is precomputed is critical. The precomputation logic adds to the circuit area and can also result in an increased clock period.

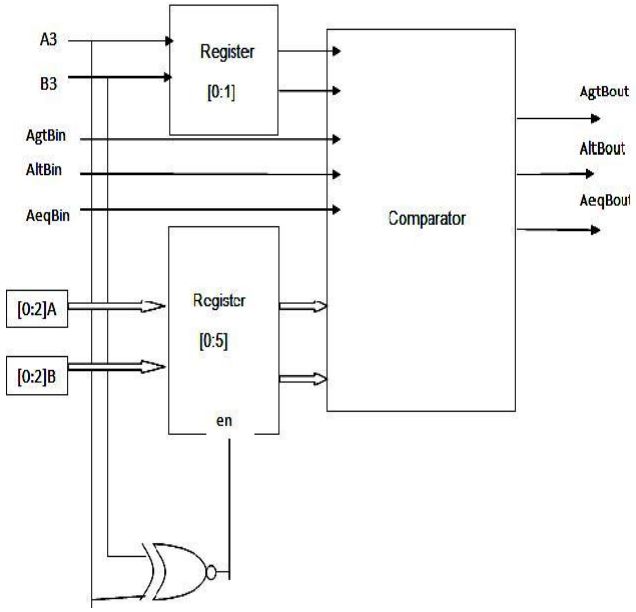

Figure5:4-bit comparator with second pre-computation architecture

6.ANALYSIS AND RESULT

When we started, implementation of 4-bit magnitude comparator in BDD, then the total product terms was 78 that means 78 node count , but with the help of BDD package tool it reduced to 46 node count. Now one node is represented by a 2x1 multiplexer. After synthesizing 2x1 multiplexer in Cadence tool, the power required for it is 29.11 nw. Since we have total 46 nodes so total power taken by 4-bit comparator is 46*29.11 nw which is equal to 1334 nw.When we synthesized 4-bit magnitude comparator in Cadence tool then the power comes as 2261.589 nw.After applying pre-computation technique in comparator then the total power comes as 1885.468 nw which is less then compare to without applying pre-computation technique in comparator. But when we compare all the three ways then we can conclude that, implementation of 4-bit magnitude comparator through BDD is the best way for low power aspect.

COMPARISON

	Cadence	Pre-computation	BDD
Power	2261.589 nw	1885.468 nw	46*29.11 nw =1334nw

7.CONCLUSION

Symbolic model checking has proven to be a powerful paradigm to automatically verify real world applications. In this thesis a BDD based optimization considering the output phase has been presented and found to achieve our objective of power minimization. The result found to be comparable to other schemes of optimization, such as pre-computation technique.In this paper I have calculated a BDD based simulation of a 4-bit magnitude comparator and compare with pre-computation strategy and find that the significant improvement in term of power.

REFERENCES

[1] S.Chaudhury and S.Chattopadhyay "Output phase assignment for area and power optimization in multi-level multi-output combinational logic circuits".

[2] B. Yang. "Optimizing Model Checking based on BDD Characterization." School of Computer Science –Carnegie Mellon University, May 1999. Available as researchreport CMU-CS-99-129.

[3] S. B. Akers, "Binary Decision Diagram," IEEE Trans. Computers, Vol. 27, 1978.

[4] K. S. Brace and R. L. Rudell and R. E. Bryant, "Efficient Implementation of a BDD Package," Design Automation Conference, 1990.

[5] P.W.C. Prasad, and A. K. Singh, "An Efficient Method for Minimization of Binary Decision Diagrams," 3rd International Conference on Advances in Strategic Technologies (ICAST), pp. 683-688, 2003.

[6] MazharAlidina, Jose Monteiro, SrinivasDevadas, "Pre-computation based sequential logic optimization for low power", IEEE transaction on VLSI system, No.4 , DECEMBER 1994.

[7] ONDREJ LHOTAK and LAURIE HENDREN, " Evaluating the Benefits of Context-Sensitive Points-to Analysis Using a BDD-Based Implementation" Sep 2008, ACM, Proceedings of the 15th International Conference on Compiler Construction. (page 19)

[8] Robert Wille and Rolf Drechsler, "BDD based synthesis of reversible logic for large function", Design automation conference,2009, DAC'09.46th ACM/IEEE.

[9] M.Morris Mano "digital logic and computer design" PRENTICE-HALL, INC., ENGLEWOOD CLIFFS, 2009

[10] Nagayama, S., A. Mishchenko, T. Sasao and J.T. Butler, 2003. "Minimization of average path length in BDDs by variable reordering".Intl. Workshop on Logic and Synthesis

[11] A. Anand Kumar "Fundamentals of Digital Design circuits", 2nd edition, PHI Learning private limited-2009

Satish Ajjappanavar received the B.E. degree in Electronics & Communication Engineering from GM institute of technology, Davanagere. At present persuing the Master of Technology in VLSI Design & Testing in BVB College of Engg & Technology,Hubli,Karnataka,India.

Dr.Anilkumar V.Nandi Working as Professor in Dept of Electronics & Communication and controller of Examination Dept in BVB College of Engg & Technology,Hubli,India